REMARKS/ARGUMENT

I. General

Claims 26 and 29-56 remain pending in the patent application. Independent claim 26 has been amended to address a non-substantive typographical error. Claims 1-25, 27 and 28 have been withdrawn from consideration.

All claims stand again rejected, now for the third time, in view of yet another reference that was available for search and examination purposes when the Application was filed. Applicants respectfully request that its past and following arguments be considered and that the rejections be withdrawn.

II. Priority Claim Under 35 USC § 119

Applicants note, with thanks, the Examiner's acknowledgment of the 35 USC Sec. 119 priority claim in this case to German patent applications 10038441.2, 10038440.4, and 10038439.0, all filed August 7, 2000. Applicants also note, however, that the Form PTOL-326 accompanying the Office Action of July 5, 2005, item 12, does not indicate such priority acknowledgment. Applicants therefore respectfully request that in the succeeding paper such acknowledgment be made of record on the Form PTOL-326 and that a Corrected Filing Receipt be issued.

III. Rejections Under 35 U.S.C. § 102(e)

Claims 26, 32, 38, 41, 43, and 49-56 stand rejected as anticipated by U.S. 5,485,620 to Sadre et al. (hereinafter Sadre). A rejection on the basis of anticipation requires each and every element of the claim, properly construed, to be identically disclosed by the single, cited reference. Applicants respectfully submit that Sadre fails to show each and every limitation of the claim.

As previously discussed in this application and correspondence with the U.S. Patent and Trademark Office, all of which is incorporated herein by reference in its entirety, the pending claims are all directed to methods for debugging programs for industrial controllers. The problems identified and addressed by the present invention are ones that touch upon industrial controller "engineering systems" versus "run-time systems," and recite limitations that are neither disclosed nor suggested by the art of record, Sadre included.

The distinction between "engineering systems" and "run time systems," which is well known in the art, is also made abundantly clear throughout the application and in particular at such places as paragraphs 0046-0048 and accompanying Figures 1and 2.

An engineering system, in brief, is used for programming an industrial controller. See, e.g., application at 0046.

A run time system, by contrast, receives programs from an engineering system and controls a "technical process" of an industrial controller. See, e.g., id.

According to an aspect of the present invention, the "run time system RTS contains a task control mechanism, which is used in debugging flowcharts." Application at 0047. This permits "the task to which a graphical element has been attached, which has been stopped by a suspend command, [to be] continued with [the] task control mechanism of the run time system." Application at para. 0015. Thus, "the task control mechanism [and accordingly the continuing of suspended commands] can be controlled by programs of the run time system." Id.

Sadre, by contrast, does not disclose anything of the sort. To the contrary, Sadre's disclosure, as relied on in rejecting claim 26 and its dependencies, states that: "Diagnostic Utility 252 enables breakpoints in Structured Text Program 188 by sending internal messages to Program Execution Task 253 and I/O Scanner 251. Program Execution Task 253 is configured by Diagnostic Utility 252 to send I/O status information to event handler 250." Whatever this passage may say, it certainly neither discloses nor suggests using a task control mechanism of a *run time system* to *continue* a suspended command.

Claim 26

Claim 26, now amended for merely typographical reasons, is directed to a method for debugging programs in industrial controllers, where graphical elements are linked using an editor in order to form a graphical flowchart. The method comprises preparing a debugging process, for an industrial control program, based on the graphical flowchart and assigning a suspend command to each graphical element. The debugging process then commences and continues until a suspend command is reached. Once the suspend command has been reached, the location of the flowchart element corresponding to the suspend command is displayed. The method then continues a task corresponding to a graphical element of the flowchart, that has been suspended by a suspend command, using a task control mechanism of the run-time system. The process then proceeds until the next suspend command is reached.

As described by the specification of the present application at paragraph 15, and as was incorporated from previously cancelled claim 27, the *task corresponding to a graphical element of the flowchart*, if suspended by a suspend command, can be continued by a task control mechanism of the *run time* system. This *run time* system is not disclosed or suggested by the diagnostic utility 252, which does not involve a run time system, but an engineering system, as those terms are understood in the field of industrial control.

Sadre neither discloses nor suggests the limitations of the amendment as claimed, nor does it provide a disclosure that would permit a worker in the field of industrial control to implement what is claimed or enjoy its advantages.

Nowhere does Sadre identify anything to do with programming of industrial controllers having a *run time system* -- as opposed to an *engineering system* -- continuing a task that has been stopped by a suspend commands.

Sadre is deficient art in yet another respect: it fails to disclose the limitation of assigning a suspend command to each graphical element of the flowchart. The language of Sadre relied upon as allegedly showing this recited feature, col. 27, lines 63-65 (claim 9) and Figure 12) not only does not disclose this limitation, but teaches away from it. In particular, the rejection alleges this recited feature is prefigured by a portion of Sadre claim 9, which depends from claim 1. But this is manifestly not the case. Read properly, and assuming without conceding that claim 1 discloses graphic elements at all, claim 9 does not disclose a suspend command being assigned to each graphic element. By incorporating the language of claim 1, claim 9 refers to *at least one* step box and *at least one* action box. If anything, this is virtually the opposite of assigning a suspend command to *each* graphic element.

Because Sadre does not describe or suggest all limitations of the invention as claimed, it cannot anticipate (nor does it suggest or render obvious) claim 26, which is therefore respectfully submitted to be allowable.